Introduction to Metagenomics

Milot Mirdita, Clovis Galiez, Johannes Söding

November 8th, 2017

1 Introduction

Microbial communities are major players of Earth's ecosystems. The study of their genomes could be of great importance not only for ecology and evolution, but also for the discovery of new potentially useful enzymes and metabolites. Most microorganisms are impossible to cultivate in laboratory conditions, but we can analyze their genome by directly sequencing samples from their natural habitats. Direct analysis of genomes contained in environmental sample is known as *metagenomics* and is a promising way to get the most information about microbial populations in specific ecosystem. Increasing usage of shotgun sequencing in metagenomics has led to rapid accumulation of data, used for discovery of new pathways, genes and species [1, 2, 3].

The typical metagenomic analysis starts with collecting the samples from an environment of interest followed by DNA extraction from these samples. Afterwards, the extracted DNA is sequenced, providing a mixture of billions of sequences (so-called *reads*, about 250bp using modern technology) over the A,T,C,G alphabet of fragments of different genomes of the microorganisms present in the sample.
The downstream analysis of these reads is only computational: quality control of the reads, assembling, annotation, etc. In typical metagenomic project, you can have hundreds of different species and tens of millions of proteins.
The purpose of this hands-on tutorial is to get familiar with modern computational tools to handle metagenomic data and get some insight into the biology behind it :)

On the Internet, there are many public resources that store metagenomic data. MG-RAST[1] is a database providing most of the metagenomics reads publicly available, together with some basic data analysis.

[1] `http://metagenomics.anl.gov`

2 Playground

We used the MG-RAST server to download four different sets of protein sequences coming from different environments. *Your mission, if you choose to accept it*, is to identify those environments using our metagenomics toolkit! Be creative![2]

Two useful tips:

- to avoid typos and gain some time: if you have partially typed a command or a file name, you can press the TAB key to get the automatic completion of your command line. If what you are typing cannot be uniquely completed, you can press the TAB key tow times to see a list of suggestions.
- to stop a program: when you want to stop a running program you can press Ctrl+C.

2.1 Working with the Shell

First, check what files you have available:

```
# places you in the data-day2 folder of your home directory
cd data-day2

# lists the files in the folder
ls

# lists the files in more detail
ls -lah

# tells you the full path to the folder you are currently situated
pwd

# goes back to the parent directory
cd ..
```

You can check the first lines contained in any of the files by running:

```
head FileName
```

The format of the sequence files is called the fasta format. It stores the sequences coming from the metagenomic datasets. The identifier of a protein is written just after the ">" symbol, and its corresponding amino-acid sequence is on the next lines.
The tsv (tab separated values) formatted files, with which you are going to work later, contain one record per line, with attributes about this record separated by "TAB" characters. This is a common representation of data in bioinformatics and easy to explore with standard linux system tools.
More commands are described in the appendix 4.1.

[2] You can later download the data yourself and reproduce the tutorial at home. On MG-RAST, there are 310,447 datasets currently (October 2017), so feel free to explore :)

2.2 MMseqs2

MMseqs2 (standing for *Many to Many Sequence Search*) is a suite of tools for searching, clustering, and filtering protein sequence sets, from single sequences to billions of sequences. This makes it perfect for using it for metagenomic analysis. You can explore its usage by running its main command in the terminal[3]:

```
mmseqs
```

2.2.1 Working with MMseqs2

You need to convert your sequence fasta files to the MMseqs2 format to be able to use the other MMseqs2 modules:

```
# creates from a fasta file a db that mmseqs can read
mmseqs createdb YourFastaFile.faa YourDBname
```

After creating the databases for each of the fasta files, you can check that the database files are actually created:

```
# lists the files in the folder
ls
```

If everything is fine, you should see four files:

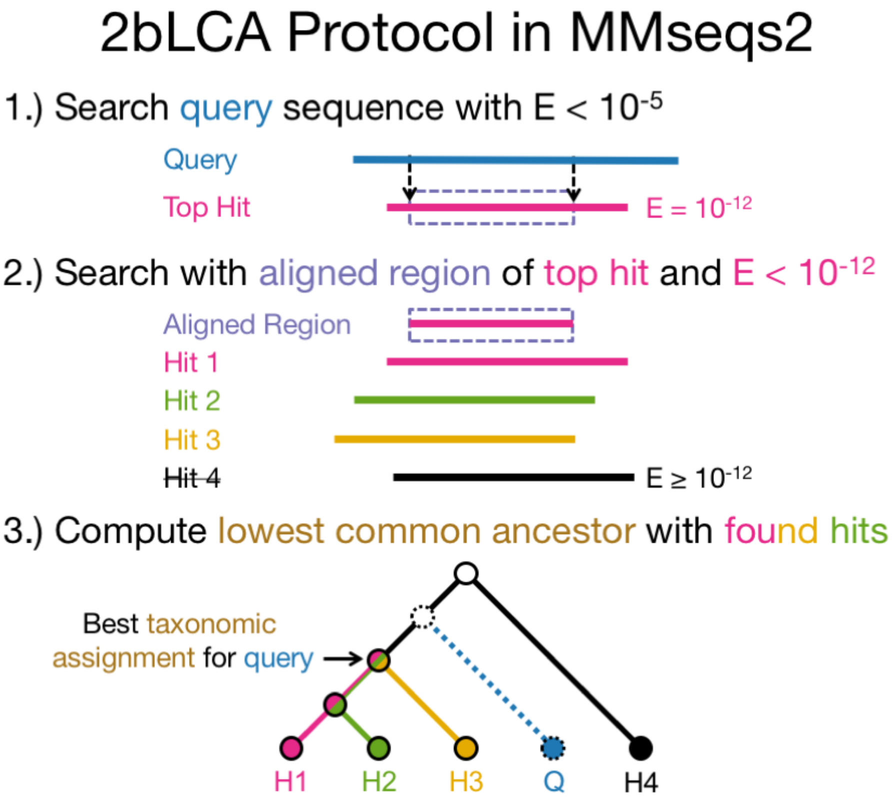
```
YourDBname
YourDBname.index
YourDBname_h
YourDBname_h.index
```

The *YourDBname_ h* file contains the header descriptor of your sequences, each separated by a NULL byte. The *YourDBname* file contains the sequences, also separated by NULL bytes. The *.index* files contain descriptions of the corresponding data file for fast access.

2.3 Identify taxonomy composition

To get insight in the ecosystem, one can identify the taxonomy of the bugs in the samples. One way to do so is to search the sequences you have freshly packed into an MMseqs2 databases against a target database of reference sequences for which we know the taxonomies. By identifying homologs through searches with taxonomy annotated reference databases, MMseqs2 can compute the lowest common ancestor. This lowest common ancestor is a robust taxonomic label for unknown sequences. By default, MMseqs2 implements the 2bLCA protocol [4] for choosing a robust LCA.

[3] To learn more about this tool, the complete MMseqs2 documentation can be found at (https://github.com/soedinglab/MMseqs2/wiki).


2bLCA Protocol in MMseqs2
1.) Search query sequence with E < 10-5
Query
Top Hit
E = 10-12
2.) Search with aligned region of top hit and E < 10-12
Aligned Region
Hit 1
Hit 2
Hit 3
~~Hit 4~~
E ≥ 10-12
3.) Compute lowest common ancestor with found hits
Best taxonomic assignment for query
H1
H2
H3
Q
H4

One of the most important manually curated reference databases is SwissProt [4][5].We have already built the necessary databases (see how in 4.2 and 4.3) and downloaded the taxonomy tree of life from the NCBI onto your computer (see 4.4). You can find the resulting database files here:

```
# lists the files in the folder
ls ~/databases/mmseqs
```

Let's get the taxonomy of the sequences contained in your database:

```
# create a tmp dir
mkdir tmp

# runs the actual taxonomy
mmseqs taxonomy YourDataBase ~/databases/mmseqs/swissclust30_2017_09_seed_db
↪  ~/databases/mmseqs/swissclust30_2017_09_seed_db_OX.tsv ~/databases/ncbi/
↪  YourTaxonomyResult tmp -s 1

# creates a Tab-separated file of the search that is human-readable
mmseqs createtsv YourDataBase YourTaxonomyResult YourTaxonomyResult.tsv
```

[4] http://www.uniprot.org/uniprot/?query=reviewed%3Ayes

[5] For a more in-depth analysis you would, for example, use our Uniclust databases [5]. These are the bigger brothers of the SwissClust30 database we prepared for you. They were built by clustering all 90 million UniProt sequences.

2.4 Looking for the molecular functions

You can identify the sequences of your databases in the reference database by running a search with MMseqs2:

```
# create a tmp dir if not existing already
mkdir tmp

# check available search parameters
mmseqs search

# run the actual search
mmseqs search YourDataBase ~/databases/mmseqs/swissclust30_2017_09_seed_db YourSearchResult
↪  tmp -s 2 -c 0.9 --min-seq-id 0.8

# creates a Tab-separated file of the search that is human-readable
mmseqs createtsv YourDataBase ~/databases/mmseqs/swissclust30_2017_09_seed_db YourSearchResult
↪  YourSearchResult.tsv
```

Can you find out what the most abundant found proteins are? Use the list of commands in the appendix[6]. Look up some of the found target sequences on the UniProt website to get a feeling of the data.

2.5 Analyzing and Visualizing our Data

We are going to look at our data with Python, Jupyter, and the ETE3-Toolkit. Start a new jupyter notebook by running in your work directory:

```
# go to your work directory
cd ~/data-day2/

# start a new notebook
jupyter notebook
```

A new browser tab should open. We prepared a basic notebook to visualize the taxa in a tree. You can download it (`taxonomy.ipynb`) on our course website[7]. Here you can see some code to turn your list of lowest common ancestors into a tree of life.

There are many databases on the web that you can use to learn more about specific species. The NCBI Taxonomy Browser (`https://www.ncbi.nlm.nih.gov/Taxonomy/Browser/wwwtax.cgi`) gathers all those resources in one place. Search for some of the most represented taxa in your tree. Wikipedia is another great taxonomy resource. Try to find out more about the organisms and their habitats in your dataset.

[6] Solution (try yourself first): *cut -f2 result.tsv | sort | uniq -c | sort -n -r | head*

[7] Shortcut on your desktop, or `http://wwwuser.gwdg.de/~compbiol/molbio_course/2017/`

3 Deep annotation with MMseqs2, example of a recently discovered virus in the human gut

If you want to explore the sensitive options of MMSeqs, let's download the genome of a recently discovered wide-spread phage, called the "crAssphage" (named after the technique used to discover it, the cross-assembly). You can search its genome on the NCBI website[8,9], and download it as a fasta file.
You can create a MMseqs2 database from the fasta file using the createdb command of MMseqs2. Then, let's extract all the open reading frames (potential proteins) of this phage:

```
# extract only potential proteins of more than 60 amino-acids:
mmseqs extractorfs yourCrassphageDB crassphageOrfs --min-length 60

# translate the ORFs to amino-acid sequences:
mmseqs translatenucs crassphageOrfs crassphageProts
```

Let's check what a sensitive search will output as results:

```
# run a sensitive search (-s 8.5, similar to BLAST) and keep distantly related
↪  homologs (e-value of 1.0: -e 1) with MMSeqs:
mmseqs search crassphageProts ~/databases/mmseqs/swissclust30_2017_09_seed_db
↪  crassPhageAnnotation tmp -s 8.5 -e 1
mmseqs createtsv crassphageProts ~/databases/mmseqs/swissclust30_2017_09_seed_db
↪  crassPhageAnnotation crassPhageAnnotation.tsv
```

You can check the different hits you get:

```
cat crassPhageAnnotation
```

What e-values do you get? Are they reliable?
You can transform the result to a TSV file and check some of the UniProt identifiers on the UniProt website. Does it make sense?
To be more sensitive, we can use iterative searches:

```
# run an iterative search:
mmseqs search crassphageProts ~/databases/mmseqs/swissclust30_2017_09_seed_db
↪  crassPhageAnnotationIterative tmp --num-iterations 3
mmseqs createtsv crassphageProts ~/databases/mmseqs/swissclust30_2017_09_seed_db
↪  crassPhageAnnotationIterative crassPhageAnnotationIterative.tsv
```

What e-values do you get? Are they more reliable? Check the UniProt annotations associated to the best e-values. Do your results make sense?

[8] https://www.ncbi.nlm.nih.gov

[9] Also on the course website

4 Appendix

4.1 Some useful Bash commands

```
# show a file inside the terminal
less myFile

# show the first ten lines of a file
head myFile

# edit a file with a graphical application
gedit myFile

# remove a file
rm myFile

# remove a folder (-r for "recursive")
rm -r myFolder

# counts lines in a file
wc -l YourFile

# Shows only the second column from a TSV file
cut -f2 YourFile

# Visualizes your file in a sorted fashion
sort YourFile

# Stores in YourFileSorted, a sorted version of your file
sort YourFile > YourFileSorted

# Shows you only unique elements in a file (the file needs to be sorted first)
uniq YourFile

# Tells you how often every unique element occurred in a file (file needs to be
↪  sorted)
uniq -c YourFile
```

You can plug the output of one command into the next one with the “|” (pipe) character:

```
# Does the sorting and counting in one step
# then sorts the list by counts
sort YourFile | uniq -c | sort -n -r
```

4.2 How we built the reference database

SwissClust30 is a clustered version of SwissClust to 30% of sequence identity, built with MMseqs2:

```
# navigate to the data folder in the home directory
mkdir -p ~/data-day2 && cd ~/data-day2

# get the swissprot database from the internet:
wget
↪  ftp://ftp.expasy.org/databases/uniprot/current_release/knowledgebase/complete/uniprot_sprot.fasta.gz
```

```
# rename it to the current release (2017_09, when we prepared this document)
mv uniprot_sprot.fasta.gz uniprot_sprot_2017_09.fasta.gz

# build the MMseqs2 database
mmseqs createdb uniprot_sprot_2017_09.fasta.gz uniprot_sprot_2017_09

# make a tmp folder
mkdir tmp

# run the clustering
mmseqs cluster uniprot_sprot_2017_09 swissclust30_2017_09 tmp --min-seq-id 0.3 --cascaded -c
↪  0.8 -s 6

# turn the clustering into a fasta file
mmseqs mergedbs swissclust30_2017_09 swissclust30_2017_09_seed uniprot_sprot_2017_09_h
↪  uniprot_sprot_2017_09 --prefixes ">"
tr -d '\000' < swissclust30_2017_09_seed > swissclust30_2017_09_seed.fasta
```

4.3 How we built the taxonomy mapping

```
# navigate to the data folder in the home directory
mkdir -p ~/databases/mmseqs && cd ~/databases/mmseqs

# get the SwissProt knowledge base from the internet:
wget
↪  ftp://ftp.expasy.org/databases/uniprot/current_release/knowledgebase/complete/uniprot_sprot.dat.gz

# turn the SwissClust30 fasta into a MMSeqs2 database
mmseqs createdb  swissclust30_2017_09_seed.fasta swissclust30_2017_09_seed_db

# map SwissClust30 IDs to Taxons
mmseqs convertkb uniprot_sprot.dat.gz swissclust30_2017_09_seed_db.mapping --kb-columns OX
↪  --mapping-file swissclust30_2017_09_seed_db.lookup

# postprocess to get a TSV file with the format SwissClust30\_ID\textbackslash{}NCBI\_Taxon
mmseqs prefixid swissclust30_2017_09_seed_db.mapping_OX
↪  swissclust30_2017_09_seed_db.mapping_OX_pref
tr -d '\000' < swissclust30_2017_09_seed_db.mapping_OX_pref > swissclust30_2017_09_OX.tsv_tmp
awk '{match($2, /=([^ ;]+)/, a); print $1"\t"a[1]; }' swissclust30_2017_09_OX.tsv_tmp >
↪  swissclust30_2017_09_OX.tsv
```

4.4 How to download the NCBI taxonomy data

```
# Create the directory for the NCBI data and navigate to it
mkdir -p ~/databases/ncbi && cd ~/databases/ncbi

# download the taxonomy data from the NCBI
wget ftp://ftp.ncbi.nlm.nih.gov/pub/taxonomy/taxdump.tar.gz
# extract
tar xzvf taxdump.tar.gz
```

References

[1] Gene W Tyson, Jarrod Chapman, Philip Hugenholtz, Eric E Allen, Rachna J Ram, Paul M Richardson, Victor V Solovyev, Edward M Rubin, Daniel S Rokhsar, and Jillian F Banfield. Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature*, 428(6978):37–43, 2004.

[2] J Craig Venter, Karin Remington, John F Heidelberg, Aaron L Halpern, Doug Rusch, Jonathan A Eisen, Dongying Wu, Ian Paulsen, Karen E Nelson, William Nelson, et al. Environmental genome shotgun sequencing of the sargasso sea. *science*, 304(5667):66–74, 2004.

[3] Yasir Bashir, Salam Pradeep Singh, and Bolin Kumar Konwar. Metagenomics: an application based perspective. *Chinese Journal of Biology*, 2014, 2014.

[4] Pascal Hingamp, Nigel Grimsley, Silvia G Acinas, Camille Clerissi, Lucie Subirana, Julie Poulain, Isabel Ferrera, Hugo Sarmento, Emilie Villar, Gipsi Lima-Mendez, Karoline Faust, Shinichi Sunagawa, Jean-Michel Claverie, Hervé Moreau, Yves Desdevises, Peer Bork, Jeroen Raes, Colomban de Vargas, Eric Karsenti, Stefanie Kandels-Lewis, Olivier Jaillon, Fabrice Not, Stéphane Pesant, Patrick Wincker, and Hiroyuki Ogata. Exploring nucleo-cytoplasmic large DNA viruses in Tara Oceans microbial metagenomes. *ISME J.*, 7(9):1678–1695, sep 2013.

[5] Milot Mirdita, Lars von den Driesch, Clovis Galiez, Maria J Martin, Johannes Söding, and Martin Steinegger. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic acids research*, 45(D1):D170–D176, 2016.